More Physics with Maple V: Differential Equations

I. Objective

To solve differential equations symbolically and numerically using the Maple symbolic solver.

II. Introduction

Many physics problems may be written as differential equations. Maple V can solve many types of differential equations symbolically. However, because not all differential equations have analytical solutions, Maple has commands that allow numerical solutions as well. We will see examples of analytical and numerical solutions in the exercises today.

As always we must critically examine the results of a Maple session to ensure that it makes sense. It is best to *assume the result is wrong until proven correct.* We will discuss methods of checking solutions in the exercises.

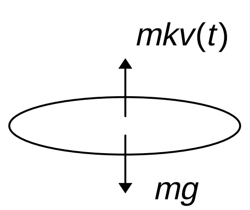
To begin a Maple session type **add maple** at the Vincent prompt, and then type **xmaple** if you are in a **X-Windows** environment.

III. Exercises

A. Symbolic Solution

Objective:	to use Maple commands to symbolically solve a differential equation
Where to begin:	start in Maple V on Project Vincent
What to do:	Follow the steps presented below.
What to turn in to your instructor:	(1) a copy of your Maple session (2) a copy of your graph
What to put in log book:	the time you begin your work, problems, solutions, new commands, etc.

(1) Skydiver: We return to a problem that you solved numerically on a spreadsheet a few weeks ago. Suppose that a skydiver falls vertically from some initial position with only the forces of gravity and air resistance acting on him/her. Suppose further that air resistance produces a drag force proportional to the sky diver's speed. Newton's second law ($ma_y = \sum F_y$) leads to the equation of motion



$$\frac{dv(t)}{dt} = g - kv(t),$$

where *v(t)* is the speed at time *t*, *g* is the acceleration of gravity, *m* is the mass of the skydiver (it occurs in the force diagram above but cancels out of the equation of motion), and *k* is the drag constant.

(2) Defining a Differential Equation: We wish now to give Maple the differential equation above. For convenience, we will call the equation **eq1**. The differential equation is written in Maple as:

eq1:=diff(v(t),t)=g-k*v(t);

and the commands have the meanings below:

Command	What it does
eq1:=	assigns the variable name **eq1** to the equation to the equation
diff(v(t),t)=g-k*v(t)	defines the differential equation of the skydiver. Note that **diff(v(t),t)** means $\frac{dv(t)}{dt}$
;	terminates command lines: tells Maple to return output

(3) Symbolic Solution: Now that the differential equation is defined, we wish to solve it. The first step is to set the initial conditions of the problem. We will assign the inital condtion **v(0)=v0** to a variable called **init** the following way:

init:=v(0)=v0;

Command	What it does
init:=	assigns the variable name **init** to the equation on the right
v(0)=v0	specifies initial condition: the value of **v** at time **t=0** is given by the variable **v0**

To solve the equation, type the following: **sol:=dsolve({eq1,init},v(t));**

The commands are explained below:

Command	What it does
sol:=	assigns the variable name **sol** to the solution of the differential equations
dsolve({eq1,init},v(t))	solves the differential equation **eq1** assuming the initial condition stored in **init** for the function **v(t)**
;	terminates command lines: tells Maple to return output

After you have entered the above, Maple should return the output

$$sol := v(t) = \frac{g}{k} + e^{(-kt)}\left(-\frac{g}{k} + v_0\right)$$

(4) Assigning Variable Names to the Solution: Though Maple has solved the differential equation, it has not permanently assigned the symbol **v(t)** to the solution above. To do so we use the command **assign** as shown below:

assign(sol);

Command	What it does
assign(sol);	assigns the solution of the differential equation stored in **sol** to the variable name **v(t)**

(5) Checking the Solution: One method of checking the result is to substitute the solution back into the original equation and see if the result is an equality. To do so type:

simplify(eq1);

Command	What it does
simplify( );	simplifies an algebraic formula
eq1	substitutes the variables found in **sol** into equation **eq1**

Maple should return an equality indicating the solution works.

(6) Checking Limiting Cases: We wish now to test the solution in several limiting cases. Two limits for which our physical intuition provides an immediate answer are: $\lim_{k \to 0} v(t)$ (the case of no air resistance) and $\lim_{t \to \infty} v(t)$ (the case of very long fall times).

For the second limit, Maple does not yet know that $k>0$, so we must give it this information. Issue the command:

assume(k>0);

Now check the limits of the solution $v(t)$. Are the results what you expect based on your physical intuition?

Maple Command	Mathematical Operation
limit(v(t),k=0);	$\lim_{k \to 0} v(t)$
limit(v(t),t=infinity);	$\lim_{t \to \infty} v(t)$

(7) 2 Dimensional Plot of the Solution: Now we want to plot the solution for $v_0,=0$, $k=1,$ and $g=9.8$ from times $t=0$ to 10 s. To do this type:

v0:=0;k:=1;g:=9.8;
plot(v(t),t=0..10);

Maple will plot the solution.

(8) 3 Dimensional Plot of the Solution (Optional): Suppose we want to study what happens to the function $v(t)$ for several different values of the drag constant k. We therefore desire to make a 3 dimension graph of the skydiver's speed as a function of time and drag constant. To do this we will establish the symbol ***V*** as a function of the independent variables **k** and **t** using the command:

k:=‘k’;
V:=unapply(v(t),k,t);

The commands have the meanings explained below:

Command	What it does
k:=‘k’;	resets k (makes it a variable, it was defined as a constant before).
V:=	assigns the variable name **V** to the right-hand side (we capitalize it to distinguish it from **v(t)**).
unapply(v(t),k,t);	defines the formula called **v(t)** as a function of two local variables called **t** and **k.**

Now we will assign numeric values to the variables $g=9.8$, $v0=0$ by typing

g:=9.8;v0:=0;

To make the 3 dimension plot of *v(t)* versus *t* and *k*, type the following:

with(plots);

plot3d(V(k,t),t=0..1,k=0..10, orientation=[-60,75],axes=NORMAL);

Command	What it does
with(plots);	loads Maples specialized plot routines
plot3d( )	initiates the 3D plot routine
V(k,t)	defines the function of two variables that will be plotted
t=0..1	defines the plot limits for the time variable as being from 0 to 1
k=0..10	defines the plot limits for the drag constant as being from 0 to 10
orientation=[-60,75]	defines the orientation of the graph (don't worrry about this)

(9) Saving Graphs: Save your graphs by going to the **File** menu on the graph window and going into the **Print** option. Select the file type **Postscript** and give your file a name. To print out the graph type the following command at the vincent prompt:

lpr -Pxxxxx -Dpostscript *filename*,

where **xxxxx** is the name of the printer.

(10) Save Sessions: Use the **Save** command under the **File** menue to save your session. Once your session is saved, you may open it later using the **Open** command in the **File** menu.

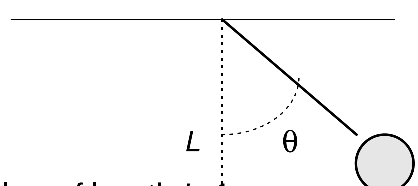
B. Numerical Solution: Uncoupled Equations

Objective:	use Maple to solve a differential equation numerically
Where to begin:	in Maple V (choose the command **New** under the **File** menu to begin a new session)
What to do:	follow the instructions below
What to turn in to your instructor:	your log book
What to put in log book:	an explanation of how you checked the solution of this problem

(1) Why Perform Numerical Solutions?: Many interesting differential equations *cannot* be solved analytically. In such cases one must turn to numerical solutions of the equations. Maple V performs numerical solutions using a fourth order Runge-Kutta Method, a method similar to the one we learned earlier when using spreadsheets.

(2) Defining the Equation and Initial Conditions: Remember from PHY 221 that if one writes Newton's Second Law for a simple pendulum, one finds that the angle θ satisfies the following second order differential equation:

$$\frac{d^2\theta}{dt^2} + \frac{g}{L} \sin\theta = 0$$



Suppose we wish to solve this differential equation for a pendulum of length L=1 m starting at rest with an initial angle of 90^0, i.e. L=1, $\theta(0) = \pi/2$ rad and $\frac{d\theta(t)}{dt} = 0$.

Because the equation has no analytical solution (recall we may only obtain an analytical solution in the "small angle approximation" when $\theta << 1$) we will solve it numerically.

We chose the initial conditions in such a way that the angles will be very large. This maximizes the differences between the real pendulum with large angles and the small-angle approximation. Because the restoring force is smaller for large angles, do we expect that the period is smaller or larger than in the small-angle approximation?

First we define the equation assigning it the name **eq1**:

eq1:=diff(theta(t),t$2)+g/L*sin(x(theta))=0;

Next we define the initial conditions and store them in a variable called **init**:

init:= theta(0)=1,D(theta)(0)=0;

Symbol	What it does
init:=	assigns the variable name **init**
theta(0)=1, D(theta)(0)=0;	defines the initial conditions of the problem, namely θ*(0)=1* and $\left.\frac{d\theta(t)}{dt}\right\|_{t=0} = 0$. Use a comma, *not a semicolon*, as separator.

(3) Solving the equation numerically: We must now assign numbers to the variables *g* and *L.* We do this using the **:=** command, and then solve the equation using **dsolve**, as shown below:

g:=9.8; L:=1;
sol:=dsolve({eq1,init},theta(t),numeric};

Symbol	What it does
sol:=	assigns the variable name **sol**
dsolve({eq1,init},theta(t), numeric)	numerically solves the differential equation **eq1** for the function **theta(t)** assuming the initail conditions in **init**

(4) Plotting the Results: One cannot use the standard **plot** routine described earlier when plotting numerical solutions of **dsolve**. Instead, we must use a routine called **odeplot**, which is called in the following manner:

with(plots);
odeplot(sol,[t,theta(t)],0..6);

We plot the result from 0 to 6 s. How many periods to you expect (roughly)?

Symbol	What it does
with(plots);	loads Maples specialized plot routines
odeplot ( );	calls a plotting routine for plotting the numerical output of **dsolve**
(sol,[t,theta(t)],0..6)	specifies that **theta(t),** stored in the numerical solution **sol,** should be plotted for the range *t=0* to 6 s.

(5) Checking the Result: Experiment with the initial conditions of the problem. Explain in your log book one method you used to check the reliability of your results.

(6) Play Around (as time permits): Compare your result with that of a simple harmonic oscillator. Are they the same? Try including the effects of air resistance to the orignal equation

$$\frac{d^2\theta}{dt^2} + \frac{k}{m}\frac{d\theta}{dt} + \frac{g}{L}\sin\theta = 0$$

Does the new solution behave as you would expect?

C. Numerical Solution: Coupled Equations

Objective:	use Maple to solve a system of two coupled differential equations numerically
Where to begin:	in Maple V (choose the command **New** under the **File** menu to begin a new session)
What to do:	follow the instructions below
What to turn in to your instructor:	your log book, graphs, and observations on what happens as you change the initial conditions
What to put in log book:	your observations

(1) Chemical Clocks: Coupled differential equations are those where two or more functions depend on the derivatives of one another. The **Brusselator** is a set of two coupled differential equations:

$$\frac{dx}{dt} = A - (B+1)x + x^2 y$$

$$\frac{dy}{dt} = Bx - x^2 y$$

These equations have been used to model the chemical clocks in cells, where x and y represent concentrations of specific compounds. The solutions $x(t)$ and $y(t)$ are roughly periodic in time. We will solve the equations for $A=1$, $B=3$, $x(0)=1$, $y(0)=0.2$.

(2) Defining the Equations and Initial Conditions: Because no analytical solution is possible (since the equations are nonlinear) we turn to numerical methods for solution. To do this we first define the equations, calling them **eq1** and **eq2**, and then store the initial conditions in the variable **init**. Type the statements below:

```
eq1:=diff(x(t),t)=A-(B+1)*x(t)+x(t)^2*y(t);
eq2:=diff(y(t),t)=B*x(t)-x(t)^2*y(t);
init:= x(0)=1,y(0)=0.2;
A:=1;B:=3;
```

(3) Solving the Equations: To numerically solve the coupled system assuming the initial conditions above, type:

```
sol:=dsolve({eq1,eq2,init },{x(t),y(t)},numeric);
```

Symbol	What it does
sol:=	assigns the name **sol**
dsolve({eq1,eq2,init},{x(t),y(t)},numeric);	numerically solves the coupled differential equations **eq1** and **eq2** for the functions **x(t)** and **y(t)** assuming the initial conditions in **init**

(4) Plotting the Solution: Study the function $x(t)$ or $y(t)$ from $t=0$ to 10. You should find that the result is periodic for long times. To plot the function, type

```
with(plots);          (if you have issued this already, skip this step)
odeplot(sol,[t,x(t)],0..10);
```

Print out a copy of your graph and submit it to your instructor.

(5) Play Around: Try changing various parameters and see what happens. The Brusselator is an example of a system with a **limit cycle**, which means that the system will settle into more or less the same long-time behavior regardless of the initial conditions that it is given. Write your activities in your log book. Do a Web search for "Brusselator" if you want to know more.

IV. Appendix 1: Basic Maple Commands

	Command	What it does
Help Command	?*topic*	gives help on a *topic*
Tutorial	tutorial( );	begins the Maple V tutorial session
Session Commands	restart;	clears all variables and begins a new session;
	save \`*sess.**m***\`	saves a Maple session to file *sess.**m*** (note that the ***.m*** extension is necessary)
	read \`*sess.**m***\`	reads a previously saved Maple session
Algebra Commands	simplify( );	simplifies algebraic expressions
	factor( );	factors algebraic expressions
	expand( );	expands algebraic expressions
	solve( );	solves specified system of equations for specified unknowns
	subs( );	substitutes specified variables into specified equations
	assign( );	assigns solutions to variable names
	evalf( );	numerically evaluates a function
Calculus Commands	diff( );	differentiates function
	int( );	integrates function
	limit( );	takes the limit of a function
	series( );	performs a series expansion of the function
	dsolve( );	solves a system of differential equations
Graphics Commands	plot( );	performs a 2D plot
	with(plots);	loads specialized plotting routines such as odeplot and plot3d
	odeplot( );	plots the numerical solution of a differentail equation
	plot3d( );	performs a 3D plot

V. Appendix 2: Saving and Printing in the X-Maple Environment

The table below gives instructions on how to perform simple operations when using **X-Maple**, the X-Windows version of Maple V.

How to Save Maple Session	pull down the **File** menu and select **Save.**
How to Open Maple Session	pull down the **File** menu and select **Open.**
How to Print Out a Maple Session	(1) pull down the **File** menu and select the **Print** option. (2) save the current session as a **Postscript** file. (3) use **lpr -P**(printer name) **-Dpostscript *filename*** command to print out file.
How to Print Out a Graph	(1) in the graph window, pull down the **File** menu; (2) choose the option **Print** and pick the format **Postscript**; (3) use **lpr -P**(printer name) **-Dpostscript *filename*** from the vincent prompt to print the file.